(\* TME 5 \*)

(\* Q.1 \*)

let rec lt\_btree (bt:'a btree) (x:'a) : bool =

match bt with

|Empty -> (1<2)

|Node(y,g,d) -> if x <=y then (1>2) else (lt\_btree g x) && (lt\_btree d x)

(\* Q.2 \*)

let rec ge\_btree (bt:'a btree) (x:'a) : bool =

match bt with

|Empty -> (1<2)

|Node(y,g,d) -> if x > y then (1>2) else (ge\_btree g x) && (ge\_btree d x)

(\* Q.3 \*)

let rec is\_abr (bt:'a btree) : bool =

match bt with

|Empty -> true

|Node (x,g,d) -> (lt\_btree g x) && (ge\_btree d x) && (is\_abr g) && (is\_abr d)

(\* Q.4 \*)

let rec mem (bt:'a btree) (x:'a) : bool =

match bt with

|Empty -> false

|Node(y,g,d) -> if (x=y) then true else if (y>x) then (mem g x) else (mem d x)

(\* Q.5 \*)

let rec insert (bt:'a btree) (x:'a) : 'a btree =

match bt with

|Empty -> Node (x,Empty, Empty)

|Node (y,g,d) -> if y>x then Node(y,(insert g x),d) else Node(y,g,(insert d x))

(\* Q.6 \*)

let abr\_of\_list (l: 'a list): 'a btree =

List.fold\_left insert Empty l

(\* Q.7 \*)

let rec list\_of\_abr (bt:'a btree) : 'a list =

match bt with

|Empty -> []

|Node(x,g,d) -> (list\_of\_abr g)@[x]@(list\_of\_abr d)

(\* Q.8 \*)

let abr\_sort (l:'a list) : 'a list =

list\_of\_abr (abr\_of\_list l)